

Bericht zum Modul Praxis III

Praxisphase: 1.1.2017 - 24.3.2017

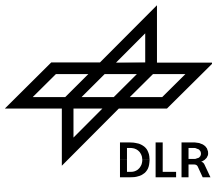
**Echtzeitfähige Analyse und Zuordnung von AIS-Signalen
zu SAR-Bildinformation**

von

Michel van Kempen

Matrikelnummer: 5036467

Kurs: TINF14ITIN



Deutsches Zentrum für
Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Standort: Oberpfaffenhofen

Microwaves and Radar Institute
Abteilung: SAR-Technologie

Betreuer: Dr. Rolf Scheiber

Eidesstattliche Erklärung

gemäß § 5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Oberpfaffenhofen, den 28. April 2017

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Abkürzungsverzeichnis	V
1 Einführung	1
1.1 Motivation	1
1.2 Aufgabenstellung	1
2 Theoretische Grundlagen der Abbildung von bewegten Zielen	3
2.1 Schiffsbewegung in Azimutrichtung	5
2.2 Schiffsbewegung in Entfernungsrichtung	6
3 Implementierung der Berechnung der Bildpositionen	7
3.1 Berechnung der Schiffssposition im SAR-Bild aus AIS-Daten	7
3.2 Umsetzung des Prototyps in Python	10
3.2.1 Eingabedaten	10
3.2.2 Programmablauf	11
3.2.3 Ausgabedaten	13
3.3 Portierung nach C/C++ und Parallelisierung	13
3.3.1 Verwendung von Cython	13
3.3.2 Aufwand für die Portierung der Module	14
4 Vergleich zwischen berechneten und simulierten Schiffsspositionen	15
4.1 Validierung des Prototyps	15
4.2 Vergleich mit Referenzposition	16
5 Zusammenfassung	20
Literaturverzeichnis	I

Abbildungsverzeichnis

1	Aufnahmegeometrie eines SAR-Bilds [1]	3
2	Verlauf der Entfernung und der Dopplerfrequenz eines fahrenden Schiffs	4
3	Abbildung von Zielen mit Bewegung parallel zur Azimutachse im Zeit-Frequenz-Diagramm	5
4	Abbildung von Zielen mit Bewegung parallel zur Entfernungssachse im Zeit-Frequenz-Diagramm	6
5	Generierung des Distanzvektors aus den AIS-Daten	8
6	Berechnung der rg/az-Koordinaten aus dem Distanzvektor	9
7	Struktur im Python-Prototyp	10
8	Datenfluss im Python-Prototyp zwischen Ein- und Ausgabe	11
9	Fehler bei Interpolation der Position (in lon/lat) eines fahrenden Schiffs mit Cubic-Splines	12
10	Vergleich zwischen simulierten und berechneten Positionen: Hafen mit ankernden Schiffen	16
11	Vergleich zwischen simulierten und berechneten Positionen: zwei schräg zur Azimutachse fahrende Schiffe	17
12	Vergleich zwischen simulierten und berechneten Positionen zweier Schiffe	17
13	Vergleich der beiden Ergebnisse der Referenzfunktion und des verwendeten Ansatzes	18
14	Ausschnitt aus QGIS: zwei Schiffe (fahrend und ankernd) annotiert mit AIS-Daten	19

Abkürzungsverzeichnis

DLR	D eutsches Z entrum für L uft- und R aumfahrt
SAR	S ynthetic A erture R adar
GPS	G lobal P ositioning S ystem
AIS	A utomatic I dentification S ystem
MMSI	M aritime M obile S ervice I dentify
JSON	J ava S cript O bject N otation
CSV	C omma- S eparated V alues
DEM	D igital E levation M odel
KML	K eyhole M arkup L anguage

1 Einführung

Diese Arbeit ist im Institut für Hochfrequenztechnik und Radarsysteme des Deutschen Zentrums für Luft-und Raumfahrt (DLR) in der Abteilung für SAR-Technologie im Rahmen der 5. Praxisphase entstanden. Die Abteilung beschäftigt sich mit der Entwicklung von Radarsystemen zur Erdbeobachtung, speziell SAR Systeme (Synthetic Aperture Radar). Der F-SAR Sensor ist die flugzeuggestützte Testplattform des Instituts. Mit ihm werden neue Verfahren zur Verarbeitung von Radardaten und Verbesserungen der Radarsensoren ausprobiert. Zudem können die Daten des F-SAR Sensors mithilfe des On-Board-Prozessors in Echtzeit verarbeitet werden. Des Weiteren ist ein zusätzlicher Empfänger für Signale des Automatic Identification System (AIS) beschafft worden, sodass in Zukunft während der Aufnahme von Radarbildern AIS-Signale von Schiffen empfangen werden können.

1.1 Motivation

Mit Radarsystemen können Aufnahmen von Gebieten unabhängig von Wetter und Tageszeit gemacht werden und sie sind daher für Überwachung und Aufklärung geeignet. Schiffe ab einer bestimmten Größe müssen für Überwachung und Navigation AIS-Nachrichten aussenden, um sich zu identifizieren sowie ihre derzeitige Position und Geschwindigkeit mitzuteilen. Durch die Zusammenführung der beiden Informationsquellen kann überwacht werden, ob das AIS verwendet wird. Verstöße können erkannt und die zusätzlichen Informationen der Schiffe können die Qualität der Schiffsignatur in den Radarbildern verbessern. Bewegte Ziele, wie Schiffe, sind in Radarbildern defokussiert. Mit zusätzlichen Informationen ließen sich die Schiffe im Bild nachträglich fokussieren. Des Weiteren würde eine gemeinsame Darstellung der beiden Informationsquellen in Krisensituationen die Interpretation der Daten vereinfachen.

1.2 Aufgabenstellung

Es wird die Zuordnung von AIS-Signalen zu ihrer räumlichen Position im Radarbild benötigt. Die Schiffsposition im SAR-Bild soll mit Informationen aus der AIS-Quelle annotiert werden können. Die Genauigkeit der Zuordnung, damit auch an engen Passagen, wie Hafeneinfahrten, zwischen einzelnen Schiffen unterschieden werden kann, muss entsprechend gegeben sein.

Im Rahmen der Praxisarbeit soll ein Prototyp in Python zur Berechnung von Bildpositionen der Schiffe aus den AIS-Daten entwickelt werden. Die Ergebnisse sollen anschließend mithilfe eines Rohdatensimulators validiert werden. Des Weiteren soll die Berechnung mit Cython parallelisiert werden und abschließend nach C/C++ portiert und in das Framework des Onboard-Prozessors eingebunden werden.

Der Bericht ist wie folgt aufgeteilt: Zuerst wird auf die grundlegende Theorie über die Abbildung von bewegten Zielen in Radarbildern eingegangen. Anschließend wird das genaue Vorgehen bei der Berechnung beschrieben, die Implementierung des Prototyps in Python vorgestellt und die Portierung nach Cython und C/C++ beschrieben. Abschließend werden die mithilfe des Prototypen erzielten Ergebnisse im letzten Kapitel validiert.

2 Theoretische Grundlagen der Abbildung von bewegten Zielen

Der SAR-Sensor fliegt entlang des abzubildenden Gebiets und tastet es mithilfe von Radarpulsen nach und nach ab. Die akquirierten Rohdaten werden anschließend fokussiert. Das entstandene SAR-Bild spannt sich zwischen Entfernungs- und Azimutachse auf. Wobei die Entfernungsachse der Blickrichtung und die Azimutachse dem Flugpfad des Sensors entsprechen. Schiffe werden im Folgenden als Punktziel modelliert. Bei der Fokussierung wird der Dopplerfrequenzverlauf aus einzelnen reflektierten Pulsen ausgewertet. Die Position eines Ziels in Azimutrichtung kann bestimmt werden, da sich die Dopplerfrequenz während des Vorbeiflugs ändert. Die Position in Entfernungsrichtung wird durch die Signallaufzeit bestimmt.

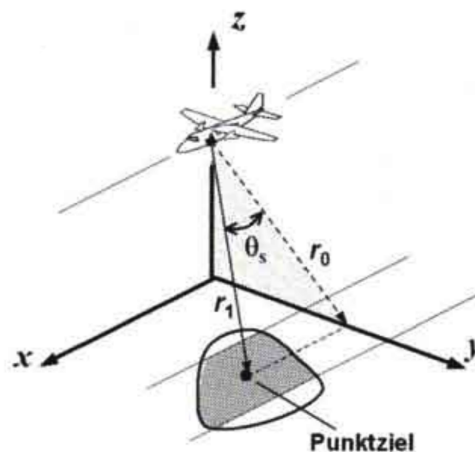


Abbildung 1: Aufnahmegeometrie eines SAR-Bilds [1]

Der Squintwinkel ist als θ_s in Abbildung 1 eingezeichnet und entspricht dem Winkel zwischen der Orthogonalen auf den Vektor der Flugbahn und dem Vektor zwischen Sensor und Hauptstrahlrichtung der Antenne. Der Winkel entsteht, wenn die Nase des Flugzeugs nicht in Flugrichtung gerichtet ist, dies kann beispielsweise durch Seitenwind verursacht werden.

Wenn sich Ziele während der Aufnahme in Bewegung befinden, verändern sie zusätzlich die gemessene Dopplerfrequenz. Daher wird das Ziel im Bild nicht an der korrekten Position auf der Azimutachse dargestellt und ist defokussiert. Dieser Versatz entlang der Azimutachse

2 Theoretische Grundlagen der Abbildung von bewegten Zielen

muss bei der Berechnung der Bildpositionen von Schiffen beachtet werden.

In Abbildung 2 ist die Entfernung zwischen Sensor und Schiff sowie der Dopplerfrequenzverlauf jeweils über Azimut, während sich das Schiff im beleuchteten Bereich befand, aufgetragen. Die Frequenz des Dopplerschwerpunkts ist im Frequenzverlauf als horizontale Linie eingezeichnet. Das Schiff wird an der Azimut-Position des Schnittpunkts der beiden Linien fokussiert (sogenannte Beam-Center-Geometrie).

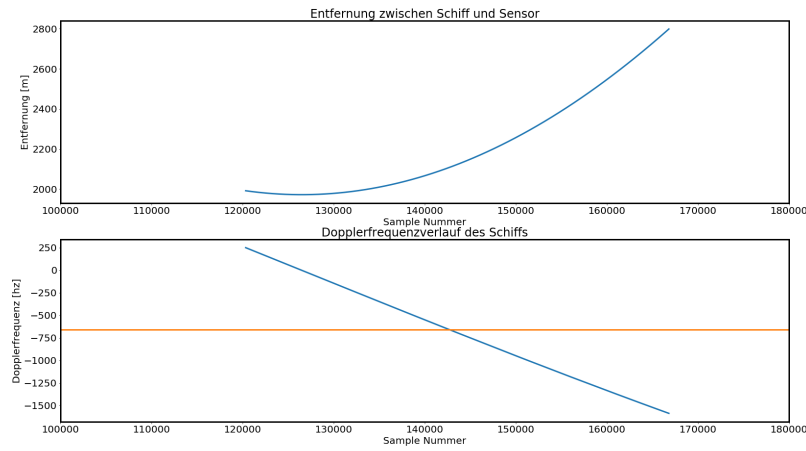


Abbildung 2: Verlauf der Entfernung und der Dopplerfrequenz eines fahrenden Schiffs

Der Doppler-Schwerpunkt f_{dc} wird nach Formel (1) berechnet und ist abhängig vom Squintwinkel θ_s . Hier ist v_0 die Geschwindigkeit des Sensors und λ die verwendete Wellenlänge. Wenn der Squintwinkel nicht vorhanden ist, befindet sich der Doppler-Schwerpunkt bei 0 Hz. Ziele werden somit an die Position fokussiert, wo sie den kürzesten Abstand zum Sensor besitzen [1].

$$f_{dc} = \frac{2 \cdot v_0 \cdot \sin(\theta_s)}{\lambda} \quad (1)$$

Die Doppler-Rate beschreibt die Steigung des Dopplerfrequenzverlaufs. r steht für die kürzeste Distanz zwischen Sensor und Flugzeug und v_{rel} bezeichnet die relative Geschwindigkeit in Azimut zwischen Sensor und Ziel.

$$k_a = \frac{2 \cdot v_{rel}^2}{\lambda \cdot r} \quad (2)$$

Die Bewegung der Schiffe kann in zwei Komponenten mit unterschiedlichem Einfluss auf den

Versatz in Azimut aufgeteilt werden. Die im Folgenden betrachteten Schiffe und das Punktziel befinden sich zum Zeitpunkt des kürzesten Abstands an derselben Position. Ausschließlich die Geschwindigkeit ist unterschiedlich. Es existiert ein Squintwinkel, sodass sich der Dopplerschwerpunkt ungleich 0 ergibt. Die gestrichelten Pfeile in den folgenden Abbildungen zeigen an, an welche Position das jeweilige Ziel fokussiert werden würde. Zudem fahren die Schiffe mit einer konstanten Geschwindigkeit. In den folgenden Diagrammen ist die Dopplerfrequenz der verschiedenen Schiffe in Abhängigkeit der Zeit schematisch dargestellt. B_a bezeichnet die prozessierte Dopplerbandbreite.

2.1 Schiffsbewegung in Azimutrichtung

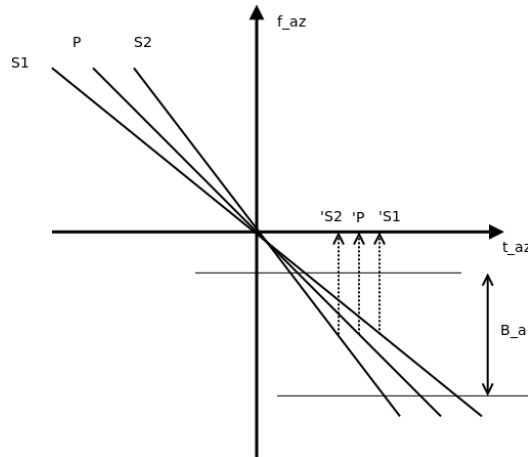


Abbildung 3: Abbildung von Zielen mit Bewegung parallel zur Azimutachse im Zeit-Frequenz-Diagramm

Die erste Komponente ist die Bewegung parallel zur Flugrichtung. Der resultierende Versatz wird in (3) beschreiben. Dabei stehen $k_{a,P}$ und $k_{a,S}$ für die Doppler-Raten von unbewegten Punktziel und Ziel, für das der Versatz berechnet wird.

$$\Delta t_{\parallel} = \frac{f_{dc}}{k_{a,P}} - \frac{f_{dc}}{k_{a,S}} \quad (3)$$

Je größer der Squintwinkel ist, desto größer ist der Versatz. Bei einem Squintwinkel gleich 0 ergibt sich keine Verschiebung für Bewegung in Azimutrichtung. In der Abbildung bewegt sich

Schiff $S1$ in Flugrichtung. Während sich Schiff $S2$ in die entgegengesetzte Richtung bewegt. Bewegung entlang der Azimutachse resultiert in einer Änderung der relativen Geschwindigkeit zwischen Sensor und Schiff und hat eine Veränderung der Doppler-Rate zur Folge.

2.2 Schiffsbewegung in Entfernungsrichtung

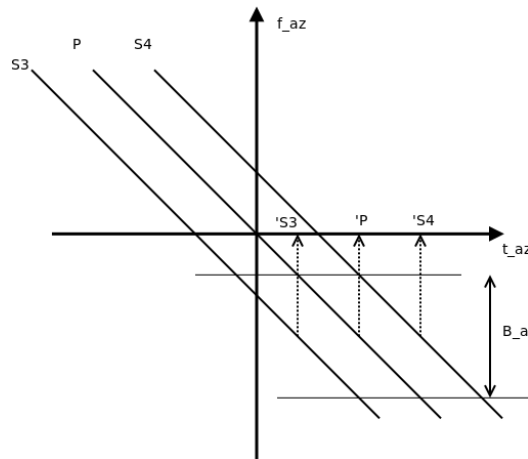


Abbildung 4: Abbildung von Zielen mit Bewegung parallel zur Entfernungsachse im Zeit-Frequenz-Diagramm

Die Bewegungskomponente parallel zur Entfernungsachse ergibt einen Versatz wie in Gleichung (5) beschrieben. Das Schiff $S4$ bewegt sich auf den Sensor zu und Schiff $S3$ bewegt sich dementsprechend vom Sensor weg. Eine konstante Geschwindigkeit parallel zur Entfernungsachse bedeutet einen konstanten Versatz in der relativen Geschwindigkeit zwischen Sensor und Schiff, sodass die Dopplerfrequenz versetzt ist. In Gleichung (4) ist θ_i der Einfallswinkel der Radarpulse auf das Ziel und v_{ship} die Geschwindigkeit des Schiffs in Entfernungsrichtung [2].

$$f_{D,mT} = \frac{2 \cdot v_{ship} \cdot \sin \theta_i}{\lambda} \quad (4)$$

$$\Delta t_{\perp} = \frac{f_{D,mT}}{k_a} \quad (5)$$

3 Implementierung der Berechnung der Bildpositionen

Das Kapitel gliedert sich in die Beschreibung des algorithmischen Vorgehens für die Berechnung der Schiffspositionen in einem F-SAR-Bild aus AIS-Daten, in die Implementierung des Prototyps in Python und in die Portierung der Funktionalität in das Onboard-Prozessor-Framework.

3.1 Berechnung der Schiffsposition im SAR-Bild aus AIS-Daten

Es sollen die Positionen der Schiffe im Bild, abgebildet durch das F-SAR, mithilfe ihrer AIS-Daten und der F-SAR-Flugparameter berechnet werden. Im Folgenden wird der Ansatz für die Berechnung beschrieben.

Die gesuchte Position eines Schiffs in Koordinatensystem des SAR-Bildes wird im Folgenden als rg/az (range/azimut) bezeichnet. Der Ausgangspunkt für die Berechnung der $[rg/az]$ -Position ist der Distanzvektor, der die Distanz zwischen SAR-Sensor und Schiff über die Zeit der Aufnahme beschreibt

Berechnung des Distanzvektors

In diesem Abschnitt wird die Berechnung des Distanzvektors d beschrieben. Diese Berechnung ist Teil einer vorherigen Arbeit und für ein besseres Verständnis hier aufgeführt.

Der Distanzvektor d_{pix} enthält für jeden Zeitpunkt, wenn ein Puls ausgesendet wurde, die Entfernung zwischen Sensor und Schiff in Pixeln bzw. Samples. Der Vektor d beschreibt die gleiche Distanz in Metern. Zuerst werden die Positionen des Schiffs aus den AIS-Daten auf den Zeitvektor des F-SAR-Tracks interpoliert. Die Positionen werden anschließend in das lokale Koordinatensystem transformiert, in dem sich auch der Flugpfad des Sensors befindet. Aus den Positionen von Sensor und Schiff wird die euklidische Distanz zueinander berechnet. Im nächsten Schritt werden alle Distanzwerte maskiert, für die sich das Schiff an diesem Zeitpunkt nicht innerhalb des beleuchteten Bereichs, bzw. innerhalb der prozessierten Dopplerbandbreite, befindet. Die Maske wird auf alle Bereiche angewendet, wo Gleichung (6) ungültig ist. In (6) beschreibt t die Zeit und PRF die Frequenz, mit der die Pulse ausgesendet

3 Implementierung der Berechnung der Bildpositionen

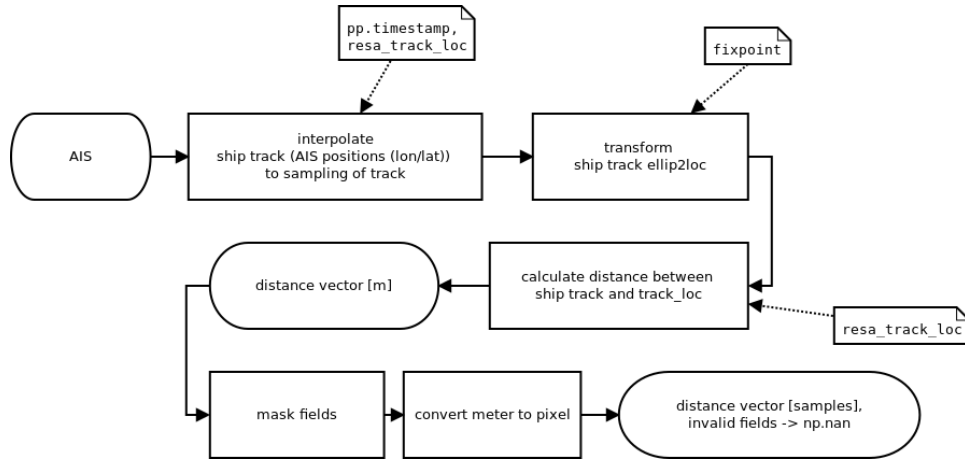


Abbildung 5: Generierung des Distanzvektors aus den AIS-Daten

wurden.

$$\frac{2 \cdot \Delta d}{\lambda \cdot \Delta t} < 0.9 \cdot \frac{PRF}{2} \quad (6)$$

Zudem werden alle Distanzen, die außerhalb der Entfernungsgrenzen der Aufzeichnung liegen, ebenfalls maskiert. Der Distanzvektor kann anschließend von Metern in Pixel umgerechnet werden. In Gleichung (7) ist r_{min} der Abstand der ersten Zeile in Entfernungsrichtung und ps_{rg} beschreibt die Größe der Pixel in Entfernungsrichtung.

$$d_{pix} = \frac{d - r_{min}}{ps_{rg}} \quad (7)$$

Berechnung der Bild-Koordinaten

Die Berechnung der rg/az Koordinaten auf Basis des Distanzvektors kann in zwei Teile gegliedert werden. Zum einen in die Berechnung der Koordinaten bezogen auf das gesamte Radarbild. Zum Anderen in die Transformation der Koordinaten in ein bestimmtes Radarbild, welches u. U. nur einen Ausschnitt des gesamten Bildes darstellt und mit einem bestimmten Faktor in Azimutrichtung neu-abgetastet (Presumming) wurde. Der gesamte Ablauf ist in Abbildung 6 gezeigt.

Ohne Berücksichtigung der Abbildungseigenschaften des SAR-Sensors in Bezug auf bewegte Ziele kann die Schiffsposition wie folgt berechnet werden: Der Zeitpunkt, an dem die Distanz

3 Implementierung der Berechnung der Bildpositionen

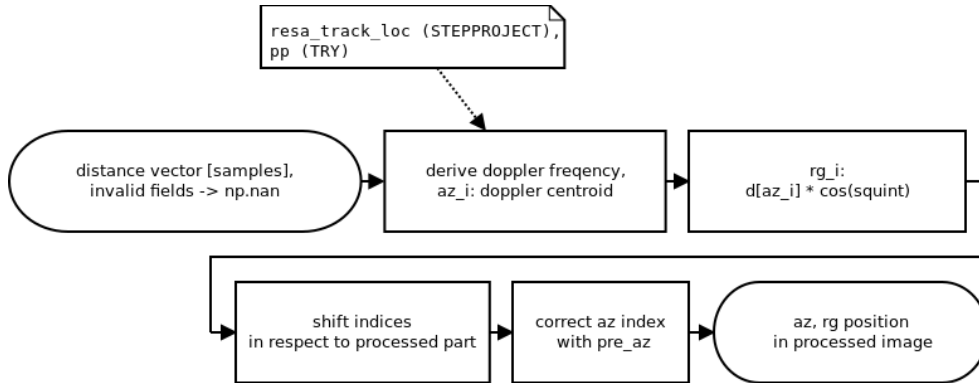


Abbildung 6: Berechnung der rg/az-Koordinaten aus dem Distanzvektor

zwischen Sensor und Schiff am geringsten ist, entspricht der Azimutposition in Doppler-Null-Geometrie. Der Wert der Entfernung zu diesem Zeitpunkt in Pixeln ist die Entfernungs-Position. Im Folgenden wird diese Berechnung als Referenz für den nächsten Ansatz mit Berücksichtigung der SAR-Abbildung verwendet. Bei der Vorstellung der Ergebnisse wird der Unterschied zwischen den beiden Ansätzen gezeigt.

Die Berechnung der Azimutposition aus gegebenen Distanzvektor d erfolgt durch die Bestimmung des Dopplerschwerpunkts in Abhängigkeit des Squintwinkels.

Zuerst wird die Dopplerfrequenz f_d für die einzelnen Zeitpunkte des Distanzvektors berechnet.

$$f_d = \frac{2 \cdot \Delta_d}{\lambda \cdot \Delta_t} \quad (8)$$

Die Position des Dopplerschwerpunkts im Vektor der Dopplerfrequenz ist die gesuchte Azimutposition.

Die gesuchte Position in Entfernungsrichtung ist r_0 , beschrieben in Gleichung (9), umgerechnet von Metern in Pixeln. In Gleichung (9) ist r der Abstand zwischen Sensor und Schiff an der zuvor berechneten Azimutposition.

$$r_0 = r \cdot \cos(\theta_s) \quad (9)$$

Im zweiten Teil der Berechnung werden die berechneten Bild-Positionen in das Gesamtbild transformiert.

3.2 Umsetzung des Prototyps in Python

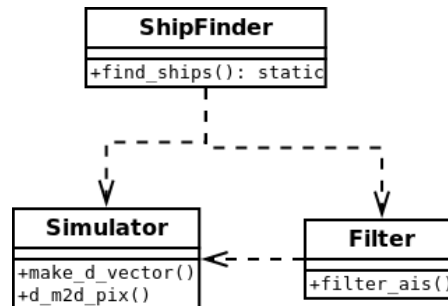


Abbildung 7: Struktur im Python-Prototyp

Der zuvor beschriebene Ansatz wurde als Prototyp in Python implementiert. Dabei wurden Methoden für das Einlesen und die Filterung von AIS-Daten sowie Methoden für die Rohdatensimulation aus bestehender Software wiederverwendet. In Abbildung 7 sind die Abhängigkeiten der einzelnen Module dargestellt. Die Funktionalität des Prototyps ist in der *ShipFinder*-Klasse zusammengefasst. Die Berechnungen verwenden die NumPy-Bibliothek und laufen nicht parallel, d.h. Schiffspositionen werden sequentiell ermittelt.

3.2.1 Eingabedaten

Die AIS-Nachrichten werden von den Schiffen ausgesendet und enthalten unter anderem die aktuelle Position. Aus den einzelnen Nachrichten wird die Schiffsposition während der Aufnahme des Radarbilds interpoliert.

Als weitere Eingangsdaten werden die Parameter des Fluges benötigt. Es werden die korrigierten Positionen des Sensors für jede Messung benötigt (*resa_track.loc*). Des Weiteren sind weitere Parameter aus den Prozessierungsparametern (pp*.xml) notwendig. Damit die Positionen, der Schiffe auf dem finalen prozessierten Bild bestimmt werden können, müssen zudem die Prozessierungsparameter des Bildes vorliegen. Wenn die Positionen der Schiffe in einem geokodierten Bild berechnet werden sollen, wird zusätzlich ein DEM für die Szene benötigt. Der Prototyp stellt eine Kommandozeilenschnittstelle zur Verfügung. Dort müssen die Pfade zu den Eingabedaten, wie in Tabelle 1 beschrieben, angegeben werden.

Tabelle 1: Eingabeparameter

AIS	AIS Datensatz
STEPPROJECT	Sensorparameter und Flugpfad
TRY	pps von finalem Radarbild
DEM	Höhenmodell für Geokodierung

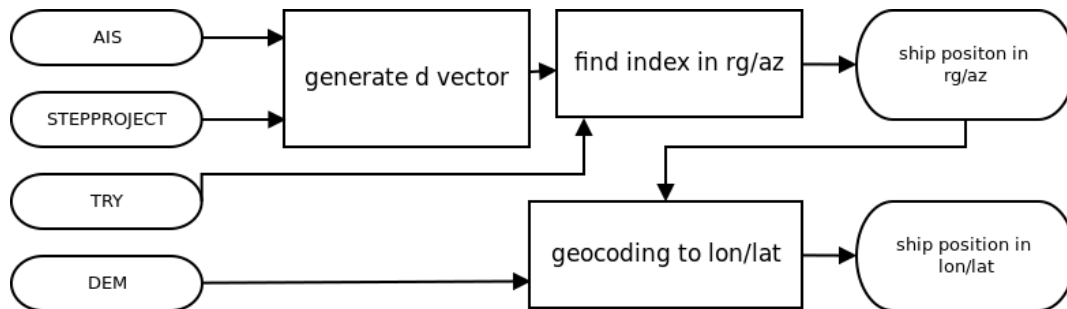


Abbildung 8: Datenfluss im Python-Prototyp zwischen Ein- und Ausgabe

3.2.2 Programmablauf

Die AIS-Daten enthalten alle Schiffe aus der näheren Umgebung. Es müssen daher zuerst die Schiffe, welche sich im abgebildeten Bereich befanden, aus dem Datensatz herausgefiltert werden. Dazu wird die Funktionalität aus dem Filtermodul verwendet. Zur Bestimmung, ob ein Schiff abgebildet ist, wird ebenfalls ein Distanzvektor zwischen Schiff und Sensor mit einer Unterabtastung von 1000 generiert.

Zudem werden pro Schiff nur die AIS-Nachrichten gebraucht, die im Zeitfenster der Aufnahme gesendet wurden. Dementsprechend werden alle AIS-Nachrichten außerhalb des Zeitfensters verworfen. Die Frequenz, mit der Schiffe AIS-Nachrichten aussenden, variiert in Abhängigkeit von Geschwindigkeit, Nähe zum Hafen usw. Zudem kann die Aufnahmezeit eines Radarbilds kürzer sein als die Pausen zwischen den einzelnen AIS-Nachrichten. Daher kommt es vor, dass keine Nachrichten innerhalb des Zeitfensters gesendet wurden. In diesem Fall werden n Nachrichten von beiden Rändern des Zeitfensters verwendet. n ist standardmäßig auf 10 gesetzt.

Da mehr Nachrichten einen höheren Aufwand bei der Interpolation bedeuten, sollte n möglichst klein gewählt sein. Die Abweichung vom idealen Ergebnis der Interpolation aus n Nachrichten ist in Tabelle 2 (experimentell ermittelt mit fahrendem Schiff) dargestellt.

Tabelle 2: Anzahl der zur Interpolation verwendeten AIS-Nachrichten und resultierende Abweichung im Ergebnis

n	durchschnittliche Abweichung vom besten Ergebnis (in Pixel)
max.	0
1	10.8
2	3.6
10	< 0.1

Für die verbleibenden Schiffe nach der Filterung werden die az/rg Positionen, wie zuvor beschrieben, berechnet.

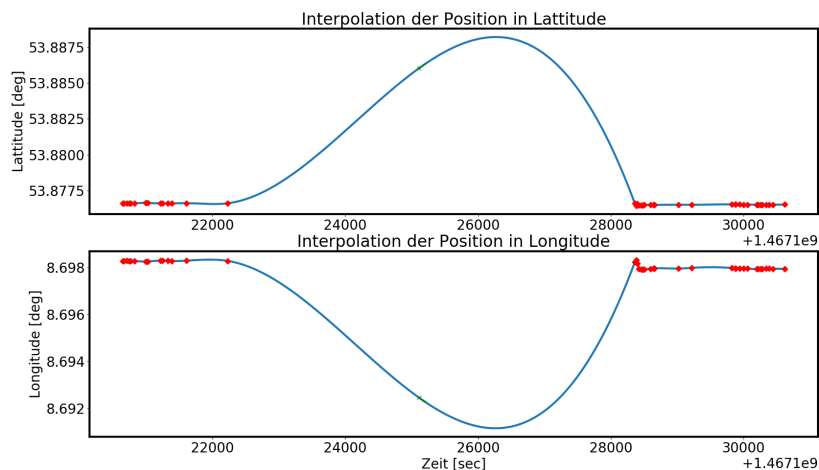


Abbildung 9: Fehler bei Interpolation der Position (in lon/lat) eines fahrenden Schiffs mit Cubic-Splines

Die Daten aus den AIS-Nachrichten werden mithilfe von Splines 3. Ordnung interpoliert. Diese Interpolationsmethode ist besser geeignet als eine lineare Interpolation, um z.B. den Positionsverlauf eines Schiffs zu interpolieren. Jedoch kann die Splineinterpolation gepaart mit AIS-Nachrichten, die mit schwankender Häufigkeit ausgesendet werden, zu extremen Interpolationsfehlern führen. In Abbildung 9 ist ein Beispiel hierfür abgebildet. Es wurde jeweils mit einer eindimensionalen Splineinterpolation die Werte von Longitude und Latitude über die Zeit interpoliert. Die Stützpunkte (rote Markierungen) sind die AIS-Nachrichten. In der Mitte im dargestellten Zeitausschnitt wurden keine Nachrichten gesendet und das

dargestellte Verhalten der Interpolation in diesem Bereich ist unerwünscht.

Dieses Problem tritt nur bei Splineinterpolationen auf. Es könnte verhindert werden, indem AIS-Nachrichten, die sich zu weit weg vom Zeitfenster der Bildaufnahme befinden, verworfen werden. Eine andere Möglichkeit ist die lineare Interpolation.

3.2.3 Ausgabedaten

Das Ergebnis ist die Schiffsposition im Bild in range/azimut (d.h. in Radargeometrie). Die Positionen der Schiffe im Bild werden in einer csv-Datei mit Schiffsname und weiteren Angaben aus dem Datenblatt der AIS-Nachrichten sowie den interpolierten Werten für Geschwindigkeit, Kurs usw. ausgegeben. Sofern die Positionen geokodiert wurden, werden sie ebenfalls in die csv-Datei oder in eine kml-Datei geschrieben.

3.3 Portierung nach C/C++ und Parallelisierung

Dieser Abschnitt soll einen Überblick über die Portierung der Funktionalität des Python-Prototyps nach C/C++ und die Parallelisierung geben. Für die Verwendung im Onboard-Prozessor in Echtzeit soll die Berechnung für jedes Schiff parallel erfolgen. Zudem muss der Python-Code hierfür nach C/C++ portiert werden. Die Portierung und Parallelisierung wurde aus Zeitmangel nicht abgeschlossen.

3.3.1 Verwendung von Cython

In diesem Abschnitt wird der Nutzen von Cython in diesem Projekt auf Basis der schon durchgeführten Portierungen analysiert. Der graduelle Umstieg mit Cython ermöglicht es Python-Code schrittweise nach C zu übersetzen. Cython kann einerseits reinen Python-Code verarbeiten. Zum anderen kann der Python-Code mit statischen Typdefinitionen usw. angereichert werden, sodass der Code vom Cython-Compiler ohne Einbindung der langsamen Python-Umgebung in C-Code übersetzt werden kann [5].

Die parallelisierten Abschnitte in Cython müssen vom Cython-Compiler in C-Code übersetzt werden können. Damit besitzt Cython für diese Abschnitte gegenüber C-Code keine besonderen Vorteile, abgesehen von der Python-ähnlichen Syntax. Die Portierung von Python-Code nach Cython mit Parallelisierung ist vergleichbar aufwendig mit der Portierung von Python nach C,

wobei C-Code ebenfalls parallelisierbar ist. Zudem ist C im Vergleich zu Cython etablierter und besser dokumentiert bzw. es sind mehr Informationen verfügbar.

Fast der gesamte Code in den zu übersetzenden Modulen soll parallel laufen, von der Filterung der AIS-Daten bis zur Positionsrechnung. Zudem muss der Code letztendlich sowieso in reinen C/C++-Code übersetzt werden. Die schrittweise Portierung ist auch in C/C++ möglich, indem einzelne Methoden in C implementiert werden, aber noch durch Python-Code aufgerufen werden. Dementsprechend hat sich bei dieser Portierung kein Vorteil durch den Zwischenschritt mit Cython gezeigt.

3.3.2 Aufwand für die Portierung der Module

Für die Berechnung wurden im Python-Prototyp Funktionen aus Modulen mehrerer Praxisphasen verwendet. Diese müssen daher zusätzlich nach C/C++ portiert werden. Zudem wurden Methoden aus Bibliotheken verwendet, für die es in C/C++ kein direktes Äquivalent gibt, wie zum Beispiel die Interpolationsroutine und die Koordinatentransformation. Die Portierung des neu entwickelten Python-Moduls Ship-Finder stellt hingegen keinen großen Aufwand da. Dafür werden aus den vorherigen Modulen Methoden für die Filterung und Generierung des Entfernungsvektors benötigt, sodass sie parallel ausgeführt werden können.

4 Vergleich zwischen berechneten und simulierten Schiffspeditionen

In diesem Kapitel sollen durch den Vergleich der berechneten Schiffspeditionen mit den Schiffspeditionen im simulierten Radarbild die Ergebnisse des Prototyps validiert werden. Zudem wird gezeigt, dass der oben beschriebene Ansatz zur Berechnung der Position notwendig ist, da der simplere Ansatz (die Referenzposition) für fahrende Schiffe nicht funktioniert.

4.1 Validierung des Prototyps

Für die Überprüfung der berechneten Positionen des Prototyps wird ein Rohdatensimulator aus einer vorherigen Arbeit verwendet. Die simulierten Rohdaten werden mit dem *STEP*-Prozessor fokussiert, sodass auf dem resultierenden Bild die Positionen der Schiffe dargestellt sind.

Beide Verfahren zur Bestimmung der Positionen von Schiffen in einem Radarbild basieren auf den gleichen Daten. Das bedeutet, dass die Simulation sowie die Berechnung auf den Positionsdaten aus dem AIS basieren. Bis einschließlich der Berechnung des Distanzvektors werden die identischen Methoden verwendet.

Die Szenen für den Vergleich bilden den Hafen von Cuxhaven mit fahrenden sowie ankernden Schiffen ab. Es werden hier Daten der VABENE-Kampagne von 2016, Flug 03 mit Pässen 02 bis 04 gezeigt.

Bei der Simulation wurden für die Schiffe jeweils Punktziele simuliert. Im Hintergrund ist in Schwarz-weiß das reale Radarbild dargestellt. Die simulierten Schiffe sind darauf in Rot überlagert. Die berechneten Bildpositionen sind durch grüne Kreise markiert.

Abbildung 10 zeigt einen Ausschnitt vom Hafen mit mehreren ankernden Schiffen. Die Positionen der Berechnung stimmen überall mit den simulierten Positionen überein. Die Markierung, welche scheinbar ein Schiff auf dem Dock anzeigt (rechts-mittig), ist durch einen Interpolationsfehler entstanden.

Auf Abbildung 11 sind zwei fahrende Schiffe abgebildet, hier stimmen die simulierten und berechneten Positionen ebenfalls überein. Die beiden Schiffe besitzen Bewegungskomponenten in beiden Richtungen. Eine mögliche Abweichung ist bei dieser Vergrößerung nicht erkennbar, sodass die Genauigkeit ausreichend ist, damit die einzelnen Schiffe den Markierungen

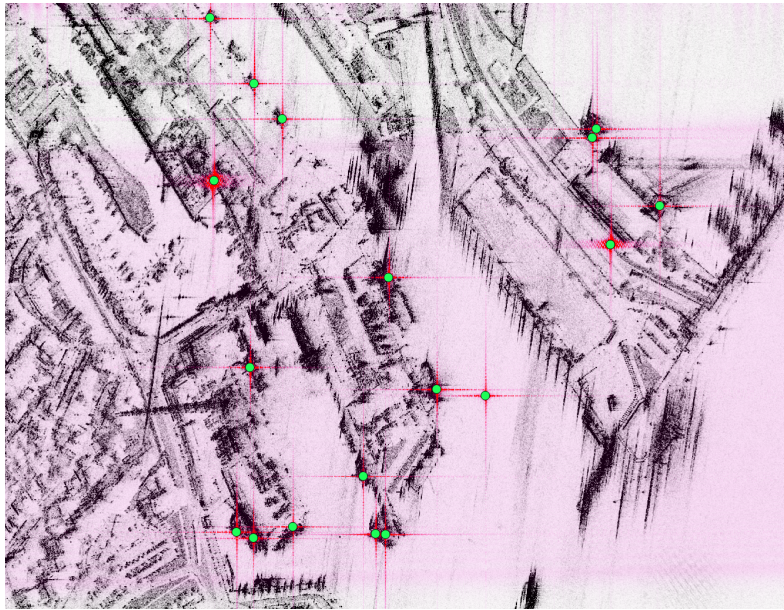


Abbildung 10: Vergleich zwischen simulierten und berechneten Positionen: Hafen mit ankernden Schiffen

zugeordnet werden können.

Im dritten Bild 12 sind die Positionen zweier Schiffe übereinstimmend markiert und simuliert worden. Das fahrende Schiff bewegt sich hier parallel zur Flugrichtung.

Wenn die SAR-Bilder und die berechneten Positionen geokodiert wurden, stimmen sie ebenfalls überein.

4.2 Vergleich mit Referenzposition

Es wird gezeigt, dass die im Kapitel der theoretischen Grundlagen beschriebene Referenzposition, welche die speziellen SAR-Abbildungseigenschaften nicht berücksichtigt, im Gegensatz zum verwendeten Ansatz nur bei unbewegten Zielen funktioniert. Dies ist in Abbildung 13 dargestellt. Die grünen Kreise markieren wie zuvor die Position des vorgestellten Ansatzes. Während die rosa Kreise die Referenzposition markieren. Die Ergebnisse der beiden Ansätze stimmen bei den ankernden Schiffen im Hafen überein. Bei den Schiffen in Bewegung ist aber, wie erwartet, ein Unterschied vorhanden und die Referenzposition stimmt dort nicht

4 Vergleich zwischen berechneten und simulierten Schiffspositionen

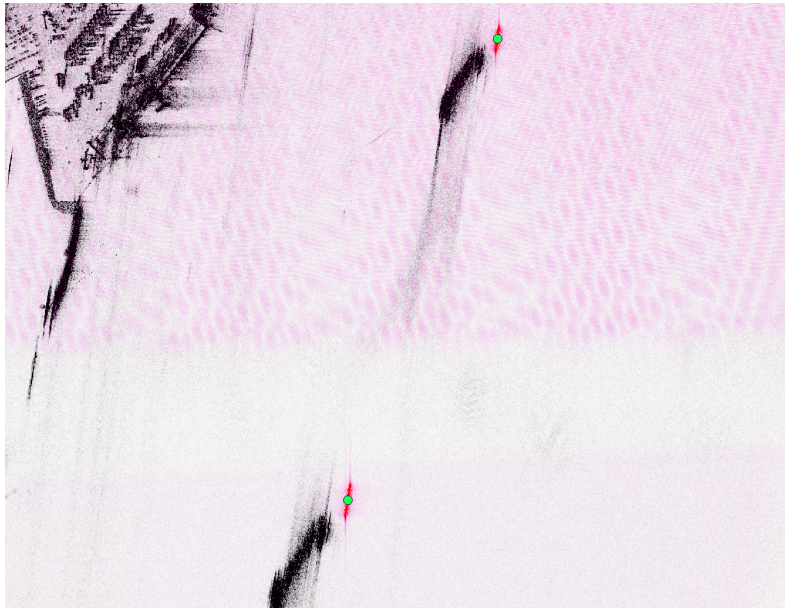


Abbildung 11: Vergleich zwischen simulierten und berechneten Positionen: zwei schräg zur Azimutachse fahrende Schiffe



Abbildung 12: Vergleich zwischen simulierten und berechneten Positionen zweier Schiffe

4 Vergleich zwischen berechneten und simulierten Schiffspeditionen

mit den simulierten Positionen überein.

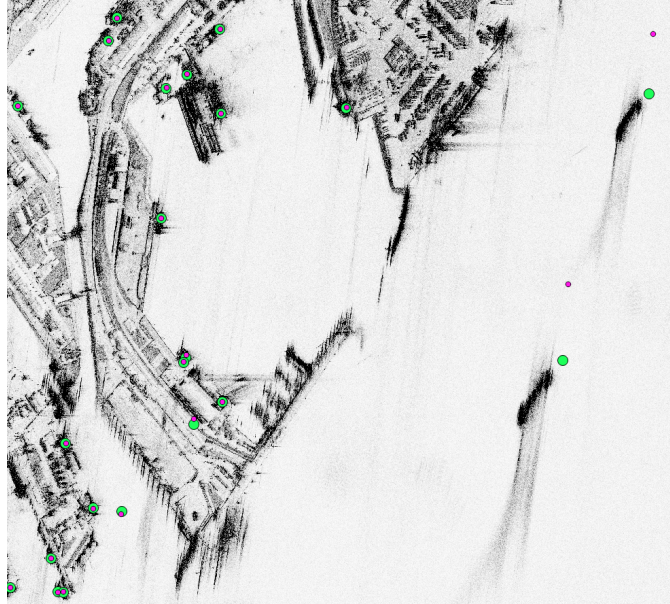


Abbildung 13: Vergleich der beiden Ergebnisse der Referenzfunktion und des verwendeten Ansatzes

In Abbildung 14 ist in einem Ausschnitt aus QGIS das reale Radarbild mit zwei Markierungen für Schiffspeditionen und annotierten AIS-Daten abgebildet. Somit sind reale SAR-Bilddaten und AIS-Daten gemeinsam dargestellt.

4 Vergleich zwischen berechneten und simulierten Schiffspositionen

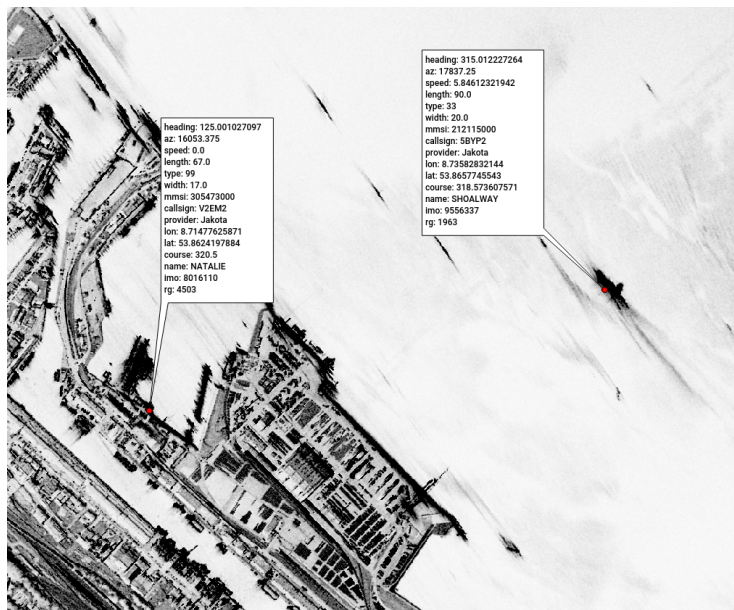


Abbildung 14: Ausschnitt aus QGIS: zwei Schiffe (fahrend und ankernd) annotiert mit AIS-Daten

5 Zusammenfassung

Es wurde ein Verfahren für die Zuordnung von AIS-Signalen zu Positionen in SAR-Bildern beschrieben. Dazu wurde in der theoretischen Betrachtung auf SAR-Abbildungseigenschaften bewegter Ziele eingegangen. Das genaue Verfahren für die Berechnung wurde, aufgezeigt und anschließend dessen Implementierung im Prototyp in Python vorgestellt. Zudem wurde die Portierung nach Cython und C/C++ beschrieben. Der Python-Prototyp ist in der Lage F-SAR Bilder mit AIS-Daten zu annotieren. Die Korrektheit der Ergebnisse wurde durch den Vergleich mit simulierten Daten bestätigt. Damit können für die, auf SAR-Bildern abgebildeten Schiffe zusätzliche Informationen aus AIS-Daten bereitgestellt werden. Diese Informationen könnten in einem weiteren Schritt für die Refokussierung der Schiffe im Bild verwendet werden.

Literaturverzeichnis

- [1] Helmut Klausing and Wolfgang Holpp. *Radar mit realer und synthetischer Apertur*. Walter de Gruyter, Berlin, 1999.
- [2] Alberto Moreira, Pau Prats-Iraola, Marwan Younis, Gerhard Krieger, Irena Hajnsek, and Konstantinos Papathanassiou. A Tutorial on Synthetic Aperture Radar. *IEEE Geoscience and Remote Sensing Magazine*, March 2013.
- [3] Russel Que, Octavio Ponce, Rolf Scheiber, and Andreas Reigber. Real-time processing of sar images for linear and nonlinear tracks. In *International Radar Symposium (IRS)*, 2016.
- [4] M. van Kempen. SAR-Rohdatensimulation von fahrenden Schiffen, Praxisbericht 4, 30 September 2016.
- [5] Website: Cython c-extensions for python, 24 April 2017. <http://cython.org/>.